Traitement d’alarmes et mise en conférence téléphonique



Travail de Bachelor réalisé en vue de l’obtention du Bachelor HES

par :

Jonathan MALFOY

Conseiller au travail de Bachelor :

Mr. Gérard INEICHEN, Enseignant HEG

Genève, le 22 octobre 2010

Haute École de Gestion de Genève (HEG-GE)

Filière Informatique de Gestion

Déclaration

Ce travail de Bachelor est réalisé dans le cadre de l'examen final de la Haute école de gestion de Genève, en vue de l'obtention d'un Bachelor en Informatique de Gestion. L'étudiant accepte, le cas échéant, la clause de confidentialité. L'utilisation des conclusions et recommandations formulées dans le travail de Bachelor, sans préjuger de leur valeur, n'engage ni la responsabilité de l'auteur, ni celle du conseiller au travail de Bachelor, du juré et de la HEG.

« J'atteste avoir réalisé seul le présent travail, sans avoir utilisé des sources autres que celles citées dans la bibliographie. »

Fait à Genève, le 22 octobre 2010

Jonathan MALFOY

Remerciements

Mr Gérard Ineichen, pour son soutien et ses conseils précieux ainsi que pour nous avoir fournis des conditions de travail optimales.

Mr Pascal Emonet, pour nous avoir mis à disposition le matériel nécessaire.

Mr Marc Boller, pour avoir répondu aux questions techniques ainsi que pour l'installation des serveurs nécessaires pour le développement de ce travail.

Mr Christophe Jordan, pour sa collaboration tout au long du travail

Résumé

L’État de Genève, plus précisément la division Réseaux/Télécoms dispose d’un système de transmission d’alarmes vocales réalisé voici une dizaine d’années. Ce système est fermé, non extensible, et dépendant du hardware qui est devenu obsolète. De plus quelques nouvelles fonctionnalités sont envisagées, afin d’étendre ce système à de multiples applications avec gestion partiellement déléguée aux utilisateurs.

C’est pourquoi il convient d’étudier et réaliser un système capable de reprendre différentes alarmes (tant des contacts secs tels que des thermostats ou des centrales de détection feu, que des éléments informatiques (réponse erronée d’une application, état snmp incorrect, traps snmp) ou encore des boîtiers IP de senseurs divers), de les traiter en fonctions de critères d’urgence et d’alarmer les bonnes personnes, en tenant compte de paramètres tels que, par exemple, les vacances des collaborateurs.

Par alarmer, il faut comprendre appeler les personnes désignées, leur jouer un texte préenregistré indiquant le type d’alarme, et leur proposer d’entrer en conférence téléphonique pour décider des mesures à prendre, ou quitter le système en cas d’impossibilité de prise en charge.

La réalisation se fera en favorisant les logiciels libres, tant du côté web (php, mysql) que du côté téléphonie (asterisk) et devra s’intégrer autant que faire se peut dans le contexte des infrastructures existantes à Réseaux/Télécoms. Le travail se déclinera en deux travaux de bachelor distincts, celui de Christophe Jordan et ce travail.

.

Table des matières

Liste des Figures

Introduction

Le traitement et la gestion d'alarmes consiste à mettre en conférence téléphonique les utilisateurs concernées lorsqu'un critère d'alarme défini préalablement se produit. Pour cela, nous avons besoin d'une interface de gestion pour les listes d'appels des personnes concernées par une alarme ainsi qu'un automate téléphonique.

Le but de ce travail est de fournir une interface web, développée en php, de gestion des listes d'appels et un système de transmission et de mise en conférence utilisant l'automate asterisk. L'utilisateur introduit à partir de cette interface web une liste d'utilisateurs à contacter en cas de déclenchement. Chaque utilisateur peut s'inscrire/désinscrire d'une liste. Le système, constitué de daemons écrit en c, doit, avertir les utilisateurs concernés par conférence téléphonique. Toutes les actions effectuées devront être journalisées.

Ce travail est divisé en plusieurs parties :

- une étude des différentes technologies utilisées ainsi qu'une explication de leur fonctionnement.
- la réalisation d'une interface web de gestion
- le développement du système de mise en conférence.

1. Asterisk et la mise en conférence

1.1 <u>Généralités</u>

Le monde de la téléphonie a considérablement évolué depuis une dizaine d'années. La principale raison de cette évolution est technique et tient à l'émergence de la téléphonie sur IP. Parallèlement, profitant notamment de l'interface IP vers la téléphonie comme canal de développement d'applications, le monde des logiciels libres de télécommunications connaît depuis la même période un essor important. Plusieurs logiciels libres ont constitué autour d'eux des communautés très actives. Parmis eux, le plus célèbre et le plus riche est sans aucun doute Asterisk.

Asterisk a pour principale fonction celle d'autocommutateur téléphonique. Cela consiste essentiellement à gérer les appels téléphoniques pour un ensemble de postes, tout comme pouvait le réaliser une opératrice voici de nombreuses années.

1.1.1 Les protocoles de la VOIP

Dans le milieu des télécoms et réseaux, chaque applicatifs a son protocole. Le plus célèbre et le plus utilisé est le protocole TCP/IP. Toute la difficulté de la VOIP est de transformer un flux voix de type audio numérique afin de le faire transiter sur les réseaux IP et ceci en temps réel.

1.1.1.1 SIP

Le protocole SIP est un protocole de transport qui date de 1996, établi par l'IETF. Il fonctionne en mode client/serveur : Le téléphone (client) envoie des requêtes au serveur SIP (autocom) ou vers d'autres clients SIP. Depuis 2004, SIP s'est imposé comme le protocole de choix en matière de VOIP en raison de sa simplicité et de sa disponibilité gratuite. Il est d'ailleurs implémenté dans la majorité des matériels et logiciels fournis à ce jour par les constructeurs et les éditeurs.

1.1.1.2 RTP

Le protocole RTP, mis au point par l'IETF en 1996, permet de transmettre tout type de données soumises aux contraintes de temps réel (audio et vidéo en particulier). RTP fonctionne sur UDP, en unicast (point à point) ou multicast (multipoint). Chaque applicatif média est transporté dans une session RTP distincte. SIP s'occupe de la signalisation et de la mise en relation de deux terminaux SIP, alors que c'est RTP qui transporte la voix.

1.1.1.3 H.323

Le protocole H.323 a été développé par l'UIT en 1996, afin de fournir un mécanisme de transport IP pour la visioconférence. H.323 est un protocole complexe qui utilise RTP pour transporter la voix, ainsi que des codecs audio et vidéo. Ce protocole est devenu obsolète. Il est préférable d'utiliser le protocole SIP qui lui, reprend les meilleurs aspects du protocole H.323.

1.1.1.4 SCCP

Le protocole SCCP est le protocole propriétaire Cisco. C'est le protocole utilisé dans tous les téléphones IP Cisco et dans l'autocom Cisco Call Manager. [1]

---

[1] Historique et définitions – Déon, S., VoIP et ToIP – Asterisk – La téléphonie sur IP

1.1.2 Plan de numérotation

Le plan de numérotation est le cœur logique du routage des appels système Asterisk : il indique au système ce qu'il faut faire à chaque fois qu'il y a un appel entrant, sortant ou interne. Défini dans le fichier *extensions.conf*, il fait appel au fichier *sip.conf* qui contient la liste des terminaux SIP du système.

1.1.2.1 Intégration avec la téléphonie de l'État de Genève

La majorité des équipements SIP de l'état de Genève sont de marque CISCO. La gestion des appels CISCO se fait à partir d'un logiciel propriétaire appelé Call Manager. Pour pouvoir utiliser ces équipements, il faut mettre en place un trunk SIP vers Asterisk dans la configuration du Call Manager. Dans ce cas-là, le fichier de configuration *sip.conf* contiendra les liens vers les différents Call Manager de l'État de Genève à la place d'une liste de terminaux SIP.

1.1.3 File d'attente

Il existe plusieurs façons d'émettre des appels depuis Asterisk à partir d'une application externe. La première est de créer un fichier texte contenant les instructions à passer à Asterisk et de copier ce fichier dans un répertoire particulier du serveur. Il tient un rôle particulier puisque Asterisk le consulte régulièrement pour récupérer ses fichiers et exécuter les instructions qu'ils contiennent.

On peut faire deux remarques sur ce type de traitement. D'abord, la fréquence de consultation du répertoire de file d'attente par Asterisk est élevée (plusieurs consultations par seconde), ce qui impose de construire le fichier dans un répertoire de travail autre, avant de copier le fichier. Sans, cette précaution, Asterisk pourrait traiter une version tronquée du fichier.

Ensuite, l'alimentation de la file d'attente d'Asterisk s'effectuant par copie ou déplacement de fichiers, il est nécessaire d'héberger l'application externe qui déclenchera les appels sur la machine qui abrite Asterisk.

Dans le cadre de l'application, la file d'attente va être utilisée pour passer les appels (un fichier par appel) ainsi que pour transmettre les différentes informations nécessaires pour une mise en conférence.

1.1.4 AMI

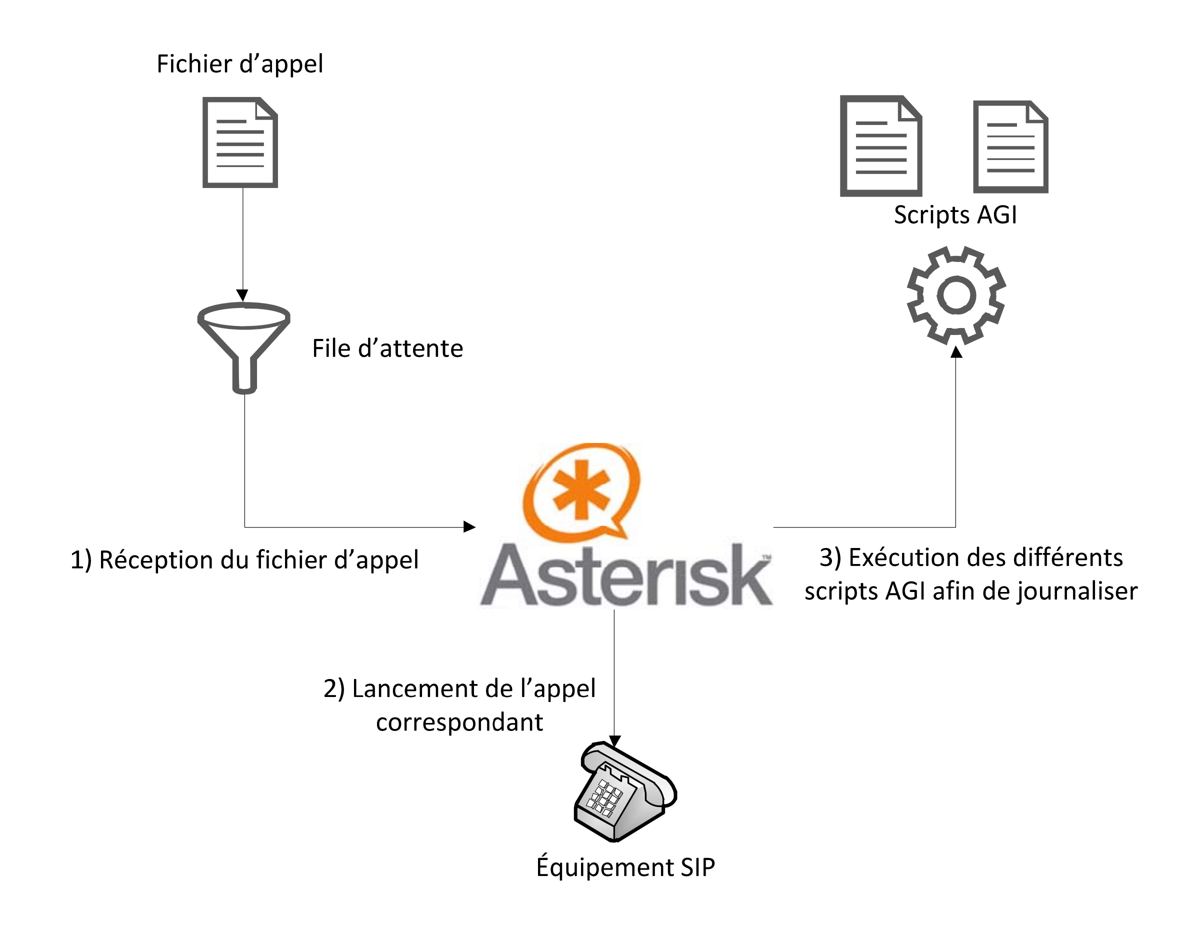
Une autre façon de passer des appels est d'utiliser l'interface AMI. AMI est une interface d'administration et de supervision d'Asterisk accessible par TCP/IP en mode client/serveur. Le principal problème d'AMI est qu'il ouvre un port sur le serveur et nécessite une sécurisation rigoureuse qui implique une restriction de l'accès TCP/IP ainsi qu'un cryptage SSL. Dans le cadre du système de mise en conférence, l'ensemble des commandes nécessaires pour passer un appel peut être utilisées dans le fichier passé à la liste d'attente. Il est donc injustifié de mettre en place un système de sécurité pour utiliser les commandes AMI alors que l'on peut utiliser le système de file d'attente de manière simple et sécurisée.

1.1.5 Script AGI

AGI (Asterisk Gateway Interface) a été développée pour Asterisk comme l'équivalent de CGI (Common Gateway Interface) pour les serveurs web. Le programme exécuté peut être écrit dans un langage quelconque, il lui suffit de pouvoir lire des informations sur l'entrée standard (stdin), écrire sur la sortie standard (stdout) et consulter des variables d'environnement. Dans le système, les scripts AGI, écrit en PERL, sont principalement utilisés pour la journalisation des actions et l'envoi des mails de notification

Figure 1

Traitement d'un appel avec Asterisk



1.2 *Mise en conférence avec MeetMe*

1.2.1 Généralités

Asterisk permet de mettre en place des conférences de façon très simple à travers une application appelée MeetMe. Chaque utilisateur interne ou externe peut donc entrer dans un espace de communication dédié. Les conférences sont mises en place à l'aide du fichier de configuration meetme.conf et d'un appel à la fonction MeetMe() dans le plan de numérotation. Il existe deux types de mise en conférence :

- Conférence statique : Un numéro fixe est donné pour chaque conférence. Les utilisateurs reliés à ce numéro seront mise en conférence. Les numéros sont enregistrés dans le fichier de configuration de l'application MeetMe. Ce type de mise en conférence ne peut pas être utilisé dans l'application car cela reviendrait à associer un numéro par alarme ce qui engendrait une modification du fichier de conférence à chaque création d'alarme.

- Conférence dynamique : Un numéro n'a pas besoin d'être enregistré dans le fichier de configuration et peut être créé à la volée pour chaque alarme. Efficace dans le cadre de cette application.

1.2.2 Conférence téléphonique

Le système doit permettre la mise en conférence les personnes concernées par une alarme. Le daemon s'occupant de la réception des alarmes[2] doit mettre en conférence les numéros de téléphone en créant un fichier d'appel dans la file d'attente d'Asterisk. L'autocommutateur va ensuite appeler les personnes et jouer un texte d'introduction spécifique selon l'alarme. L'utilisateur devra confirmer son entrée en conférence par l'appui d'une touche spécifique. Il rejoindra alors les autres utilisateurs dans la conférence dédiée. À tout moment de la conférence, les personnes pourront émettre une quittance d'alarme une fois le problème traité. Dès qu'une quittance a été émise, la conférence est automatiquement terminée ce qui met fin aux appels en cours et un mail de récapitulation est envoyé pour chaque personne désirant être notifiée. De plus, dès qu'un utilisateur décide de rentrer en conférence, une notification vocale est jouée indiquant à l'appelant le nombre d'utilisateurs déjà présent. Un bug dans Asterisk ne permet pas de jouer ce message en français lorsqu'on passe par un fichier d'appel[3].

---

[2] Voir chapitre 3

[3] https://issues.asterisk.org/view.php?id=17673

1.2.2.1 Comportement des utilisateurs en conférence

Il existe plusieurs cas de figure où il est nécessaire de relancer une conférence. Par exemple, dans le cas où les utilisateurs ne répondent pas, l'application doit les rappeler selon un critère temps défini pour l'alarme en tenant compte du cas d'urgence.

1.2.2.2 L'inconvénient de la Boite vocale

Une redirection vers la boite vocale d'un téléphone est considérée comme une réponse à un appel par Asterisk. La confirmation de l'entrée en conférence est donc nécessaire afin d'éviter la mise en conférence de la boite vocale de l'utilisateur.

1.2.2.3 Envoi de confirmation en codes DTMF

Un code DTMF (dual-tone multi-frequency) est une combinaison de fréquences utilisée pour la téléphonie moderne. Techniquement, chaque touche d'un téléphone correspond à un couple de deux fréquences audibles qui sont jouées simultanément. De cette façon, huit fréquences bien distinctes permettent de coder seize touches[4]. Dans le cadre de la conférence, lorsque l'utilisateur appuie sur une touche pour rentrer en conférence ou pour quittancer l'alarme, un code DTMF est envoyé. Il existe plusieurs façons de capter un signal DTMF sur l'Asterisk. Les principaux modes sont « RFC2833 » et « Inband ». Le mode « RFC2833 » est la norme actuelle pour l'envoi de signal DTMF. Elle est implémentée dans la majorité des Natels actuels. Malheureusement lors des tests effectués avec différentes marques, il s'est avéré que certains mobiles de marque HTC[5] ne prenaient pas encore en compte cette norme. Il n'y a actuellement pas de solution à ce problème externe venant du constructeur.

---

[4] Définition - http://fr.wikipedia.org/wiki/Dtmf

[5] Tests effectués sur les appareils HTC suivants: HTC Touch HD, HTC HD2 et HTC Desire ainsi que les OS Windows mobile 6.1, Windows mobile 6.5, Android 2.1, Android 2.2

1.2.3 Texte à jouer

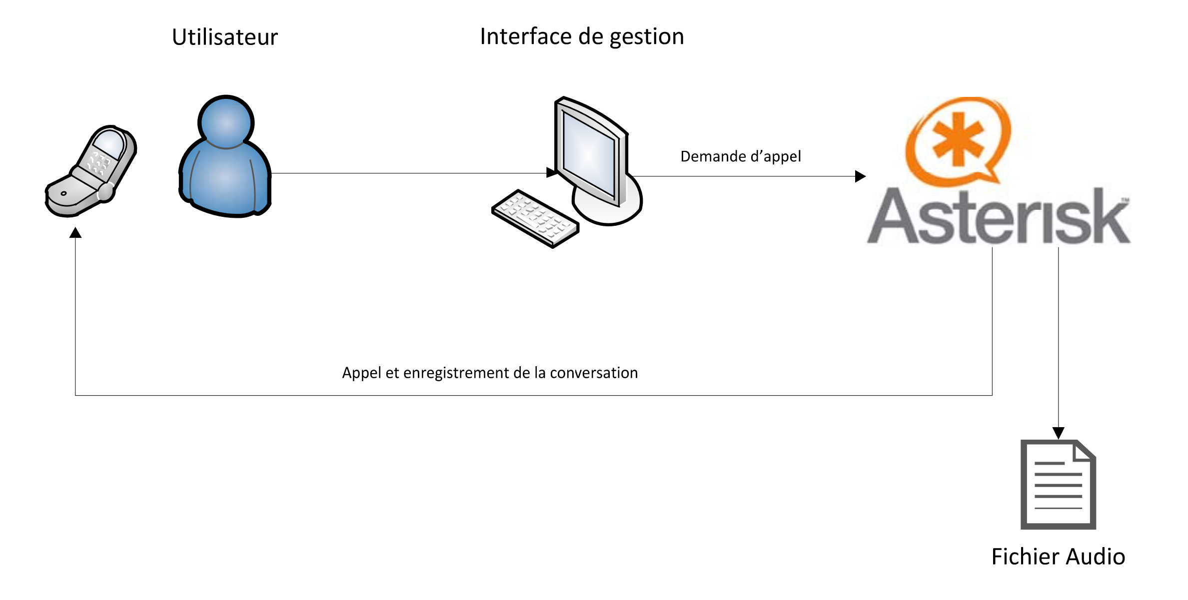
Pour chaque type d'alarme, un texte personnalisé devra être joué avant la mise en conférence afin d'avertir l'utilisateur sur la nature de l'alerte. Asterisk est capable de jouer du texte lors d'un appel en passant par l'application Playback prenant en charge des fichiers wav. Pour l'enregistrement des textes, on peut bien sûr créer soi-même les fichiers audio à partir d'une application externe mais Asterisk propose deux façons d'enregistrement rapide et efficace.

1.2.3.1 Enregistrement vocal

L'application Record d'Asterisk permet d'enregistrer un texte directement depuis le téléphone. L'utilisateur peut appeler un numéro spécial dédié à l'enregistrement ou le système peut utiliser la file d'attente ce qui va permettre à Asterisk d'appeler la personne pour lui proposer un enregistrement. L'avantage des enregistrements vocaux est que l'utilisateur va pouvoir directement mettre en paramètre, à partir du site web de la gestion des listes d'appel[6], l'enregistrement associé à l'alarme.

Figure 2

Enregistrement vocal



[6] Voir chapitre 2

1.2.3.2 *Text-To-Speech*

L'Université d'Edimbourg a développé un moteur de traduction Text-To-Speech appelé Festival. Cette application open-source permet de transformer un texte en parole et d'effectuer une lecture audio à l'interlocuteur au bout de la ligne. Malheureusement, à l'heure actuelle, Festival ne propose pas encore de voix française. Dans le système, l'implémentation du Text-To-Speech est intéressante pour effectuer une lecture par défaut des noms d'alarmes lorsqu'aucune phrase n'a été préenregistrée par l'utilisateur.

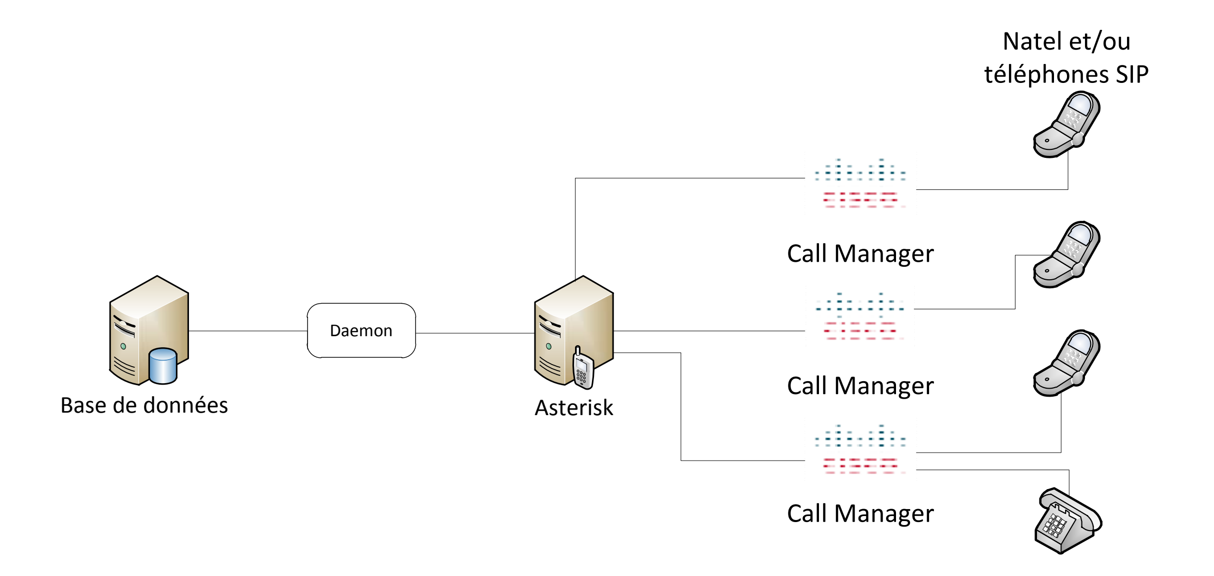
1.3 ***Configuration et architecture d'Asterisk pour le traitement d'alarme***

Le système de mise en conférence se divise en trois modules.

- Une base de données qui contient les différentes informations sur la conférence telle que les liens vers les enregistrements vocaux d'intro et de fin pour indiquer la nature de la conférence ainsi que la liste des numéros d'appel.
- Un Asterisk contenant un plan de numérotation configuré avec un trunk sur les Call Manager de l'état de Genève. La mise en conférence est configurée par défaut pour recevoir des variables externes ainsi que divers scriptes AGI.
- Un daemon jouant le rôle d'intermédiaire afin d'éviter une dépendance entre la base de données et Asterisk[7].

Figure 3

Architecture de mise en conférence



[7] Voir chapitre 3

1.3.1 Les informations nécessaires pour la mise en conférence

Plusieurs informations, stockées dans la base de données, sont nécessaires à la configuration d'une conférence téléphonique:

- Un enregistrement vocal d'introduction à la conférence contenant la nature de l'alarme ainsi qu'un rappel des commandes DTMF disponible. Par exemple – « *Bonjour, une alarme porte du bâtiment C a été déclenchée. Appuyez sur # pour rentrer en conférence. Pour quittancer l'alarme, vous pouvez appuyer à tout moment sur ** »
- Un enregistrement vocal de fin de conférence manuelle pour notifier les utilisateurs. Par exemple - « *L'alarme a été quittancée, la conférence est maintenant terminée* »
- Un enregistrement vocal de fin de conférence afin de notifier les utilisateurs que la conférence a pris fin.
- Un numéro de conférence qui sera automatiquement généré par Asterisk. C'est le principe de conférence dynamique.
- Une liste des numéros de téléphone à appeler.
- Une ou plusieurs adresses email afin de notifier l'utilisateur des actions prises pendant le déroulement du traitement de l'alarme.
- Les informations à journaliser (type d'alarme, heure, etc…)

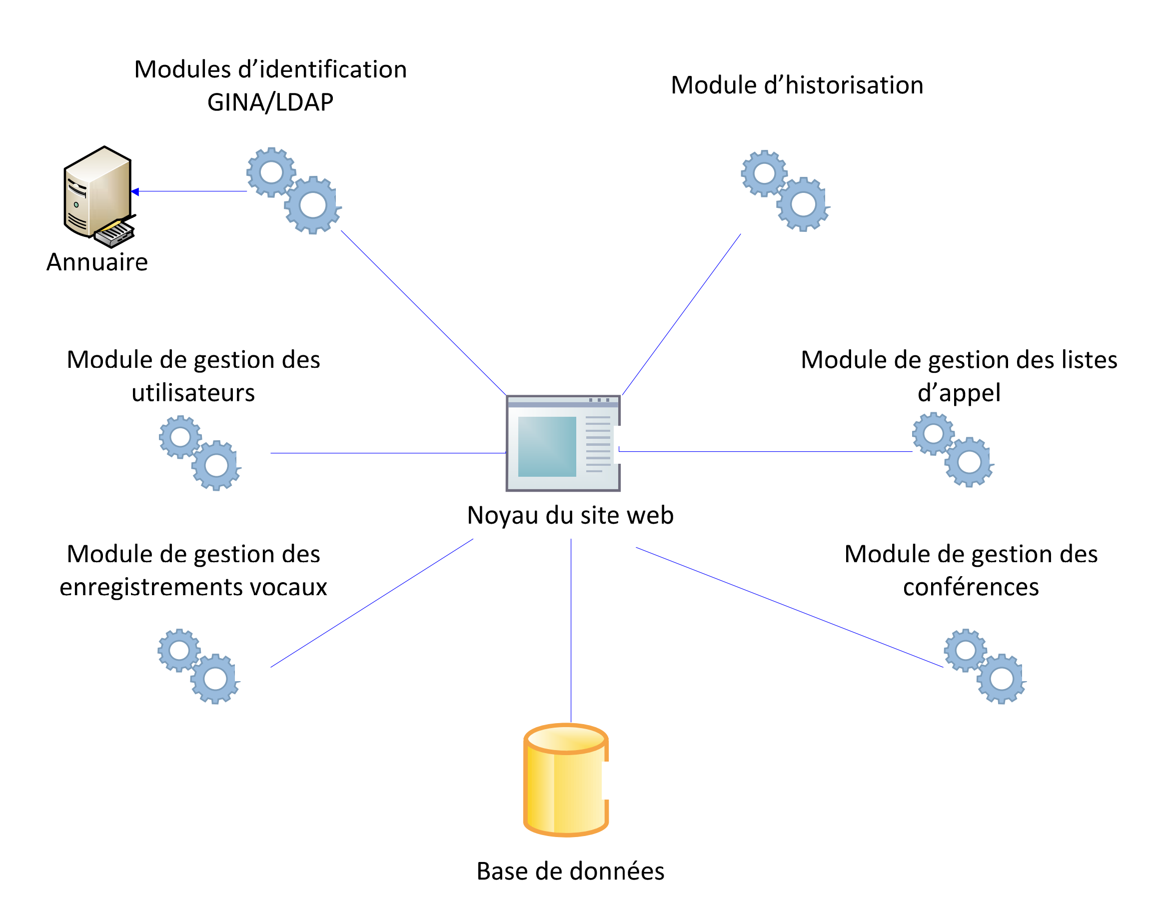
2. Gestion des listes d'appel

2.1 *Généralités*

Les utilisateurs concernés par les déclenchements d'alarmes peuvent depuis une interface web ajouter, modifier ou supprimer les différentes configurations de mise en conférence selon leurs droits. L'accès au site web ainsi que les rôles sont attribués depuis l'authentification GINA ou LDAP. Cette interface de gestion web a été développée à partir d'un template de la division Réseaux/Télécoms du Centre des Technologies de l'Information de l'état de Genève afin d'avoir une base commune et facilement maintenable puisque déjà existante.

Figure 4

Architecture globale du site web



2.2 *Les interfaces*

2.2.1 Gestion des listes d'appel

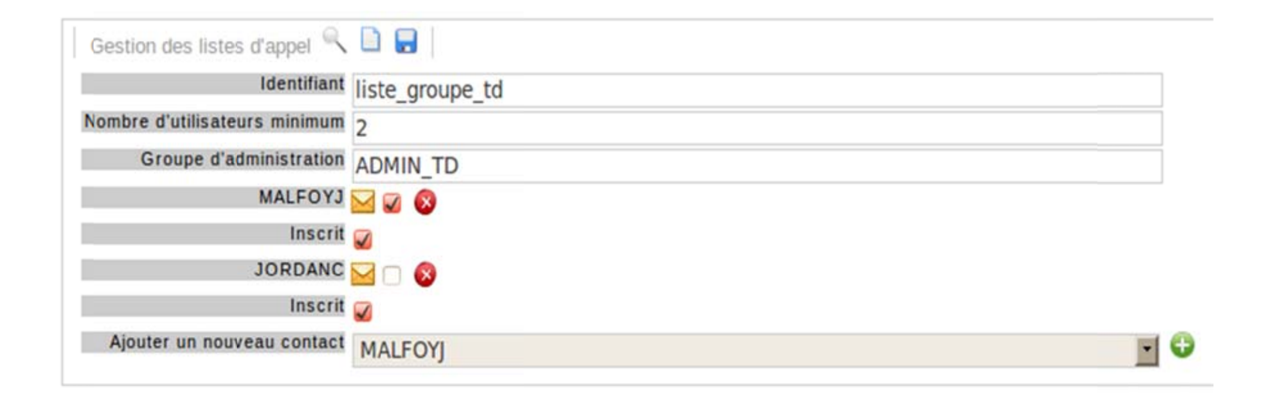
Une liste d'appel doit contenir :

- Un identifiant
- Un nombre d'utilisateurs minimum inscrit à la liste afin d'éviter un trop grand nombre de désinscription qui aurait pour conséquence la mise en conférence d'un seul utilisateur ou pire, aucune personne.
- Un groupe d'administration autorisé à modifier la liste et les paramètres de mise en conférence[8]
- Une liste d'utilisateurs avec la possibilité pour chaque personne de s'inscrire/désinscrire de la liste d'appel ainsi qu'une option de notification par email pour les utilisateurs voulant recevoir un résumé de traitement d'une alarme

Une liste d'appel a été intentionnellement séparée d'une conférence car un utilisateur doit pouvoir associer plusieurs fois la même liste pour plusieurs conférences.

Figure 5

Interface de gestion des listes d'appel



[8] Voir chapitre 2.3

2.2.2 Gestion des utilisateurs

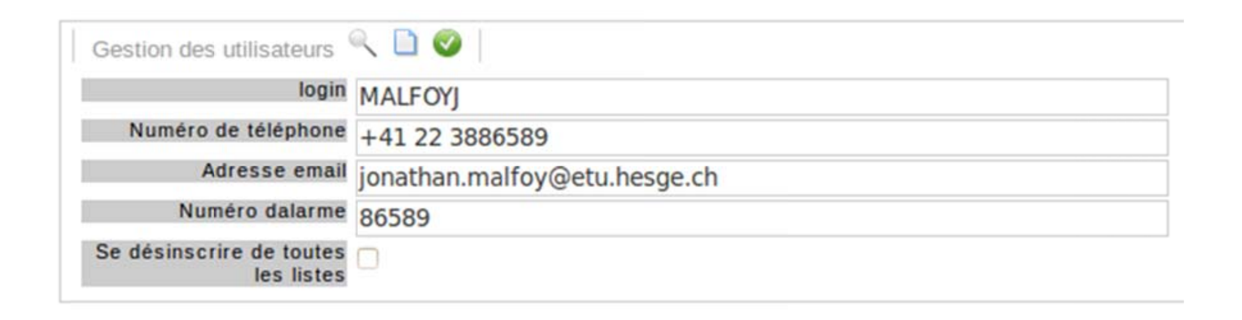
Cette section permet l'administration des utilisateurs de l'application. Elle contient les données propres à chaque utilisateur :

- Un login généré lors de la première connexion par GINA ou LDAP ou inséré manuellement par l'administrateur.
- Un numéro de téléphone professionnel
- Une adresse email pour permettre la réception des notifications d'alarme
- Un numéro d'alarme à appeler en cas de déclenchement d'alarme. Ce numéro est libre, il peut être relié à tout appareil dont les Natels.
- La possibilité de se désinscrire de toutes les listes d'appel lors, par exemple, d'un départ en vacances.

Chaque utilisateur peut modifier ses propres informations si besoin est.

Figure 6

Interface de gestion des utilisateurs



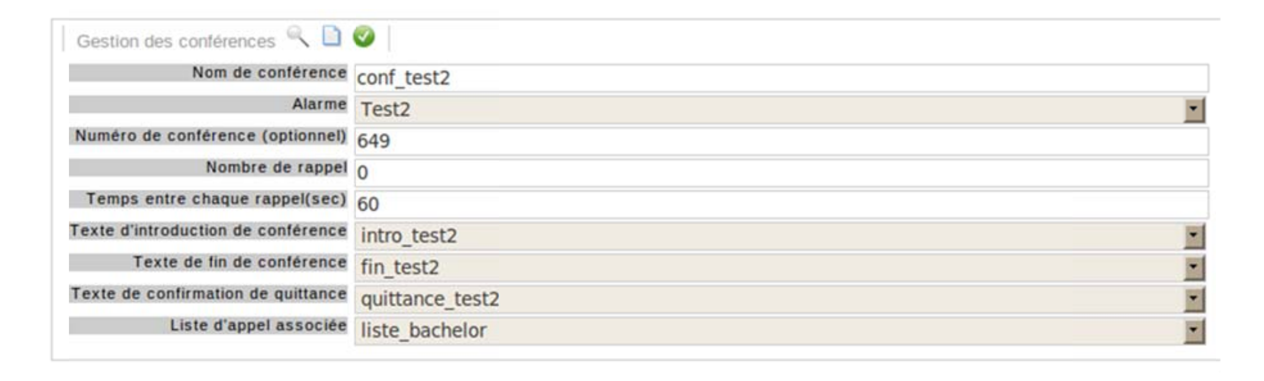
2.2.3 Gestion des conférences

L'interface de gestion des conférences contient tous les paramètres nécessaires pour la mise en conférence par Asterisk. Elle contient :

- Un nom de conférence servant d'identifiant pour l'utilisateur
- Un numéro de conférence. Si ce champ n'est pas rempli, l'application va attribuer un numéro de conférence libre de façon aléatoire (principe de conférence dynamique d'Asterisk)
- L'alarme associée à la conférence. La liste des alarmes disponibles est générée à partir d'une table externe à la base de données. Les alarmes sont administrées à partir de l'interface de gestion des critères[9].
- La possibilité de rappeler un certain nombre de fois l'utilisateur qui ne décroche pas selon le cas d'urgence ainsi que la durée entre chaque rappel.
- Les textes à jouer d'introduction, de fin de conférence ainsi que la confirmation de quittance de l'alarme.
- La liste d'appel associée à la conférence

Figure 7

Interface de gestion des conférences



[9] Voir chapitre 3

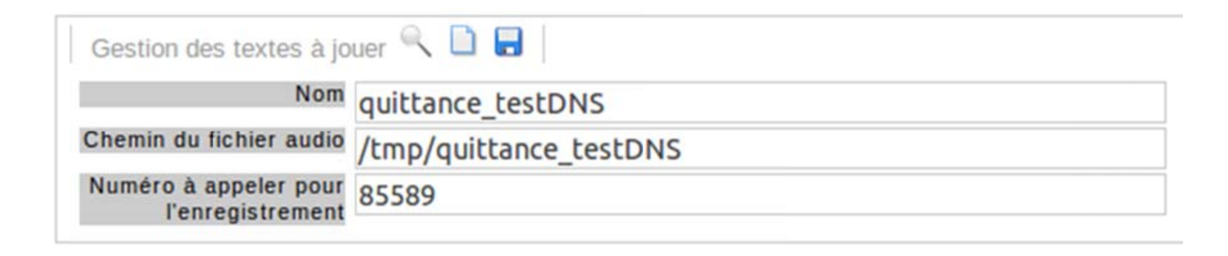
2.2.4 Gestions des textes à jouer

L'utilisateur peut enregistrer des textes directement depuis son téléphone. En effet, afin d'éviter l'enregistrement depuis un programme tiers ainsi qu'un envoi de fichier sur le serveur, l'interface se servira d'Asterisk pour l'enregistrement de textes vocaux grâce au système d'appel depuis la file d'attente. L'utilisateur entre son numéro de téléphone et reçoit instantanément un appel pour l'enregistrement. Les informations nécessaires sont :

- Un identifiant pour permettre l'association dans la gestion des conférences.
- Un chemin sur le serveur pour l'enregistrement du fichier.
- Le numéro à appeler pour l'enregistrement. Ce numéro est complètement libre.

Figure 8

Interface de gestion des textes à jouer



2.3 Authentification

L'ensemble du système de gestion est restreint aux ayant droits. A la première connexion, l'utilisateur doit s'authentifier au moyen d'un protocole défini selon son type de compte (GINA ou LDAP)

2.3.1 GINA

Gina est un protocole d'authentification propriétaire de l'État de Genève. L'utilisateur est redirigé vers une page dédiée et s'authentifie à l'aide d'un couple login/mot de passe de son compte GINA.

Figure 9

Page d'authentification GINA



2.3.2 LDAP

LDAP (Lightweight Directory Access Protocol) est un protocole permettant l'interrogation et la modification des services d'annuaire. Dans le cadre de l'application de gestion, LDAP va authentifier l'utilisateur avec le service d'annuaire Active Directory. A la connexion, l'utilisateur doit rentrer son login (préfixé de son domaine, par exemple, « GE-EM ») et son mot de passe à partir d'une simple fenêtre de connexion html.

Figure 10

Fenêtre d'authentification LDAP



2.4 Droits d’accès

Les opérations d’ajout, de suppression et de modification des informations de chaque module de l’interface de gestion ne doivent pas être attribués à n’importe quel utilisateur authentifié. C’est pourquoi un système de gestion par rôle et groupes a été introduit. Les rôles et les groupes auquel appartient chaque utilisateur est récupéré dans l’annuaire du compte de l’utilisateur.

2.4.1 Les différents rôles

L’application peut être administrée selon deux rôles :

- **L’administrateur** : L’utilisateur possédant les droits d’administrateur a le contrôle total de l’interface. Il peut non seulement, effectuer des ajouts, suppression et modifications sur l’ensemble des données, mais aussi voir en tout temps l’historique d’utilisation des utilisateurs.
- **L’utilisateur de base** : Un utilisateur de base est limité à ses propres informations, c’est-à-dire la modification de ses données personnelles comme son numéro de téléphone et son email. Il peut aussi s’inscrire ou se désinscrire des listes auxquelles il appartient et décider de recevoir les mails de notification d’alarme.

2.4.2 La gestion par groupe

Chaque liste d’appel possède un groupe d’administration. Lors de la création d’une liste, l’administrateur peut spécifier un groupe d’utilisateurs ayant les droits d’administration sur la liste. Chaque utilisateur faisant partie de ce groupe pourra modifier la liste. La modification reste restreinte. Le groupe d’administration ne pourra pas supprimer la liste. Le principal avantage de déléguer les droits d’administration sur les listes est que l’utilisateur n’aura pas à faire appel à l’administrateur global à chaque ajout dans une liste.

3. Traitement des alarmes à transmettre

3.1 Généralités

Le traitement des alarmes est le cœur du système. Il se décompose en trois parties :

- La détection des déclenchements d'alarmes ainsi que la récupération des informations correspondantes. Le programme « Daemon de traitement » a été spécialement conçu pour cette tâche.
- La mise en conférence avec la liste d'appel associée à l'alarme. Ce travail est réalisé à partir d'Asterisk.
- La quittance de l'alarme et l'envoi des notifications. Le programme « Daemon de quittance » est chargé d'effectuer ce travail.

3.1.1 Interaction avec le travail de Christophe Jordan

C'est ici que se fait le lien avec le travail de Christophe Jordan[10]. Ce travail possède une base de données contenant les informations sur les alarmes. Deux tables sont mises en commun pour effectuer un traitement d'alarme :

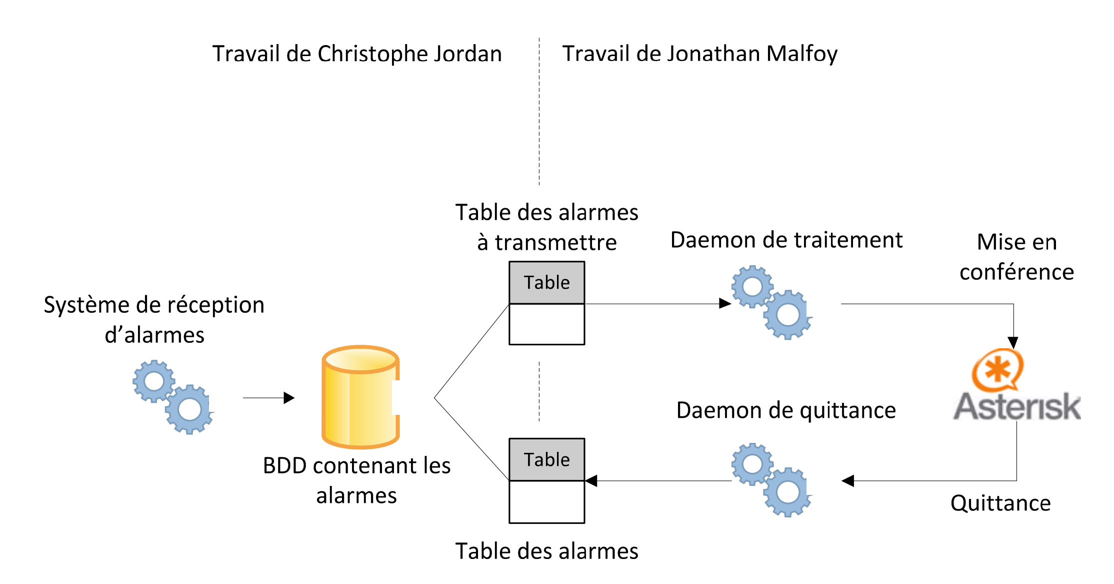
- Une table de traitement où sera inséré un enregistrement lors d'un déclenchement d'alarme contenant le lien vers l'alarme en question.
- Une table nécessaire pour notifier la quittance à l'application de Christophe Jordan.

---

[10] Voir travail de Bachelor de Christophe Jordan « Réception et transmission des alarmes sur un réseau IP » pour plus de détails.

Figure 11

Interaction entre les deux travaux



3.1.2 Définition d'un daemon

Un daemon désigne un type de programme informatique, un processus qui s'exécute en arrière-plan plutôt que sous le contrôle direct d'un utilisateur. Les daemons sont souvent démarrés lors du chargement du système d'exploitation, et servent en général à répondre à des requêtes du réseau, à l'activité du matériel ou à d'autres programmes en exécutant certaines tâches[11].

---

[11] Définition - http://fr.wikipedia.org/wiki/Daemon

3.2 Daemon de traitement

Ce programme joue le rôle d'intermédiaire entre les bases de données et Asterisk. Écrit en C, Il tourne en continu sur la machine hôte. Sa principale tâche est de détecter le déclenchement d'une alarme, de récupérer les informations de mise en conférence associé à l'alarme et d'utiliser Asterisk pour appeler les personnes concernées. Outre sa fonction de traitement, il journalise aussi tous les échanges.

Figure 12

Architecture du Daemon de traitement

Données de mise
en conférence

Daemon de traitement

Asterisk

Données des alarmes

Journalisation

3.2.1 Récupération des alarmes

Lors d'un déclenchement d'alarme, une entrée est créée dans une table, spécialement conçue pour le traitement, dans la base de données du travail de Christophe Jordan. Le daemon va alors procéder en trois temps.

- Premièrement, lorsqu'un enregistrement est inséré dans la table des alarmes à transmettre, le daemon va récupérer les informations nécessaires sur l'alarme dans la base de données. Une fois l'enregistrement récupéré, le daemon va mettre un flag pour indiquer que l'alarme est en cours de traitement.
- Deuxièmement, il va chercher les informations de mise en conférence (liste d'appel et configuration de conférence) liées à cette alarme.
- Troisièmement, le daemon va transmettre ces informations à l'Asterisk qui va s'occuper d'appeler les personnes concernées.
- Finalement, si les utilisateurs appelés ont quittancé l'alarme, le daemon va supprimer l'enregistrement et indiquer dans la base de données que l'alarme a été quittancée. Si l'alarme n'a pas été quittancée en conférence, la daemon va retransmettre les informations de mise en conférence à Asterisk dans un laps de temps prédéfini par l'administrateur.

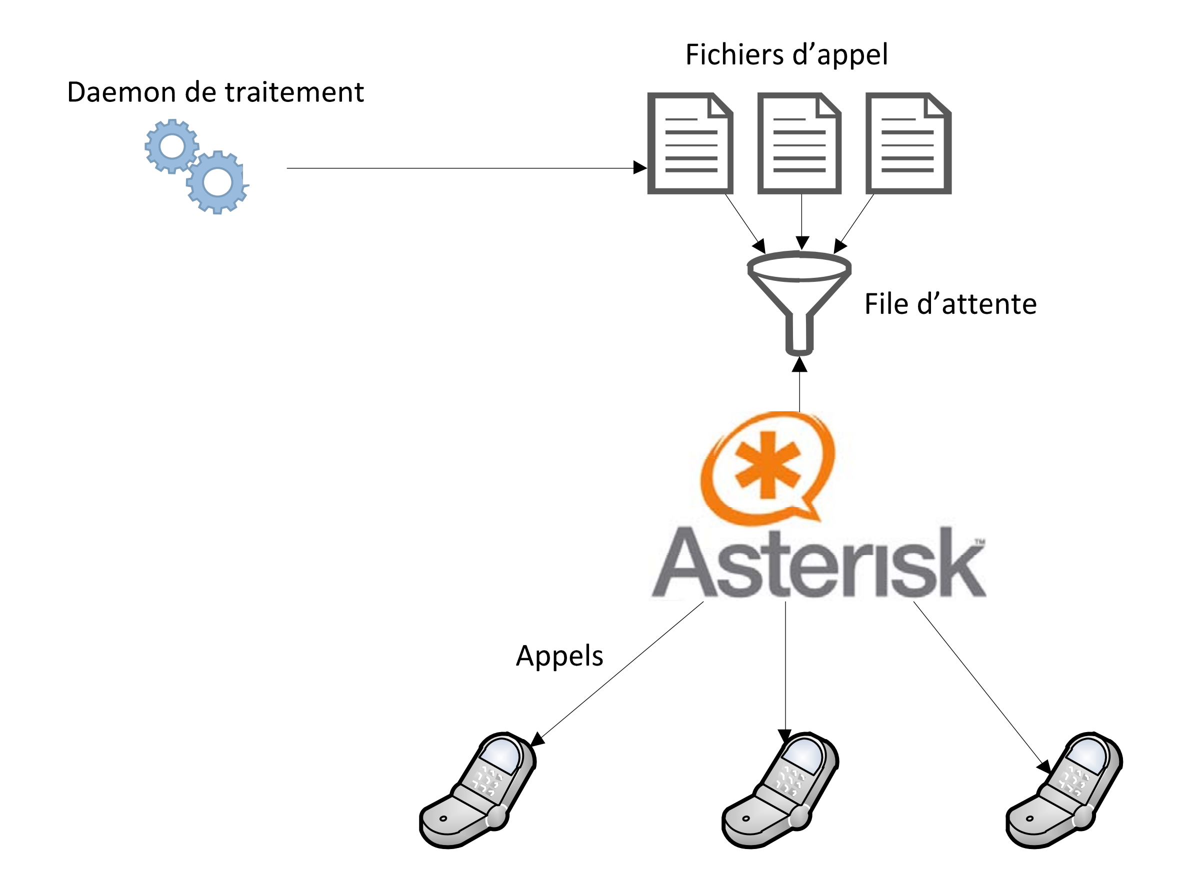
Il se peut qu'une personne traite l'alarme avant d'être mis en conférence. Par exemple si la personne se trouve à côté de l'appareil ayant déclenché l'alarme. Dans ce cas-là, l'application de Christophe Jordan supprimera l'enregistrement dans la table des alarmes ce qui évitera une mise en conférence inutile.

3.2.2 Interfaçage avec Asterisk

La communication Daemon-Asterisk se fait essentiellement à partir du principe de file d'attente. Dès que le daemon aura toutes les informations nécessaires pour mettre en conférence les utilisateurs d'une liste d'appel, il va créer un fichier par appel contenant le numéro de téléphone à appeler ainsi que des données permettant de journaliser les événements.

Figure 13

Interface avec Asterisk



3.3 Daemon de quittance

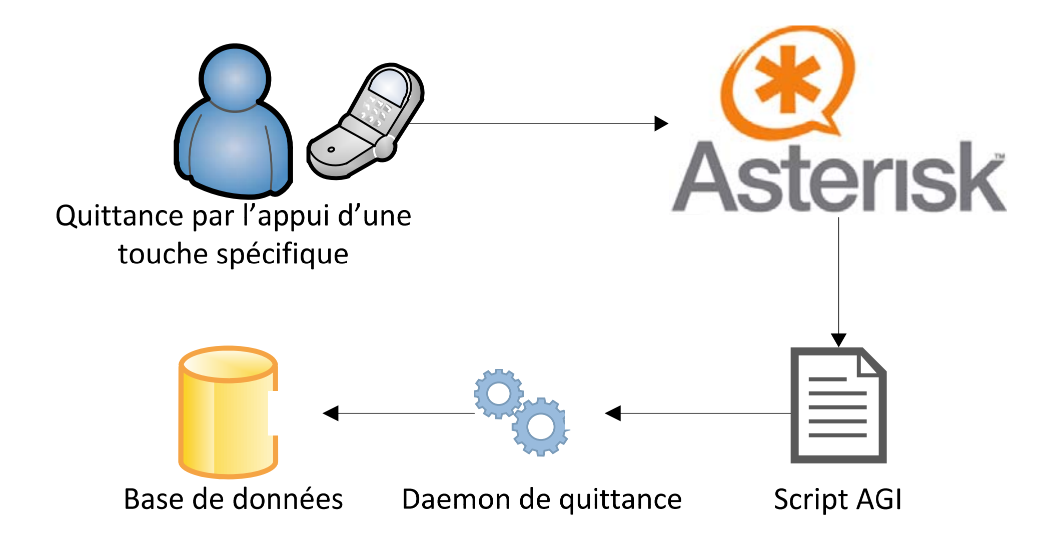
Le daemon de quittance effectue la liaison entre les scripts AGI d'Asterisk et la base de données. Lorsqu'un utilisateur va émettre une quittance pendant une conférence, la daemon va réceptionner cette quittance et notifier la base de données que l'alarme a été traitée oralement.

3.3.1 Réception et transmission d'une quittance

Comme mentionnée dans le chapitre 1, Asterisk peut faire appel à des scripts AGI. Lors d'une conférence, lorsqu'un utilisateur décide de quittancer l'alarme par l'appui d'une touche, un signal DTMF est émis et réceptionné par Asterisk. Il va ensuite faire appel à un script AGI spécialement conçu pour réceptionner la quittance. Le script va alors communiquer (à travers un socket) au daemon que l'alarme est quittancée. Les scripts sont écrits en PERL car les librairies Asterisk pour ce langage sont déjà préinstallées et maintenue par la communauté open-source.

Figure 14

Fonctionnement de la transmission des quittances



4. Journalisation complète des actions

Les alarmes nécessitent un traitement rigoureux de la journalisation. En effet, si une panne réseau ou une erreur système devait venir interférer le processus de traitement et de mise en conférence, les conséquences pourrait être relativement importantes selon la nature de l'alarme touchée. C'est pourquoi l'administrateur doit pouvoir retrouver facilement la faille ainsi que l'alarme en cause. Le moyen le plus sûr et le plus fiable est d'utiliser le protocole SYSLOG, déjà présent sur les systèmes unix.

4.1 Protocole SYSLOG

Syslog est un protocole définissant un service de journaux d'événements d'un système informatique. C'est aussi le nom du format qui permet ces échanges. En tant que protocole, Syslog se compose d'une partie cliente et d'une partie serveur. La partie cliente émet les informations sur le réseau, via le port UDP 514. Les serveurs collectent l'information et se chargent de créer les journaux.

L'intérêt de Syslog est donc de centraliser les journaux d'événements, permettant de repérer plus rapidement et efficacement les défaillances d'ordinateurs présents sur un réseau.[12]

4.1.1 L'application RSYSLOG

Rsyslog est un daemon de journalisation de type syslogd ayant pour fonctionnalités principales le support de MySQL, syslog/tcp, RFC 3195, les listes d'expéditeurs autorisés, le filtrage sur n'importe quelle partie du message et un contrôle très fin du format de sortie[13]

---

[12] Définition - http://fr.wikipedia.org/wiki/Syslog

[13] Définition - http://wiki.monitoring-fr.org/integration/rsyslog

4.1.2 Formatage des informations pour l'application

Afin de mieux traiter et trier la journalisation de l'application, la mise en place d'un formatage spécifique pour les messages a été appliquée sur l'ensemble des données journalisée. L'application utilise trois types de messages syslog :

- INFO : Les messages de ce type sont utilisés pour la journalisation des évènements faisant partie du processus normal du système. Par exemple : Une mise en conférence produira les messages suivants :

  « [TRAITEMENT] Nouvelle conférence pour alarme : TestDNS »

  « [TRAITEMENT] Appel de : 0793420000 pour alarme : TestDNS »

  « [TRAITEMENT] Appel de : 0766840000 pour alarme : TestDNS »

  « [TRAITEMENT] Conférence établie pour alarme : TestDNS »

- WARNING : Ces messages sont générés lorsqu'un évènement sortant du déroulement normal de l'application a été produit comme, par exemple, une perte temporaire de la connexion mysql. Un évènement de type WARNING ne quitte pas le daemon, c'est-à-dire qu'aucun message SIGKILL ou SIGTERM ne sera envoyé.

- ERROR : Ces événements se déclarent lorsqu'un disfonctionnement a été détecté dans le système qui pourrait nuire au bon déroulement de l'application ainsi qu'à l'intégrité des données. En général, un évènement ERROR met fin au daemon (SIGKILL ou SIGTERM).

Tous ces messages syslog sont préfixés par deux mots-clefs.

- [INIT] : Les évènements se produisent lors d'une initialisation. Par exemple, lors de la récupération des informations de mise en conférence.

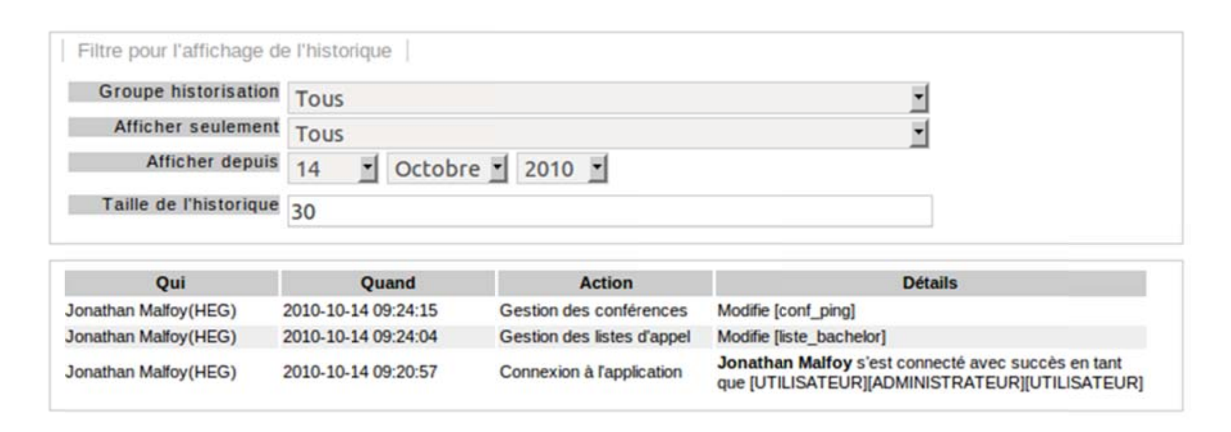
- [TRAITEMENT] : Les évènements se produisent pendant le traitement d'une alarme. Par exemple, lorsqu'un utilisateur quittance.

4.2 *L’interface de gestion*

Toutes les connexions d’utilisateurs, les modifications, les ajouts ainsi que les suppressions sont journalisés dans une table de la base de données. L’administrateur global à accès à ces informations depuis l’interface de gestion à partir d’un onglet appelé « historique ».

Figure 15

Historique de l’interface de gestion



Qui	Quand	Action	Détails
Jonathan Malfoy(HEG)	2010-10-14 09:24:15	Gestion des conférences	Modifie [conf_ping]
Jonathan Malfoy(HEG)	2010-10-14 09:24:04	Gestion des listes d'appel	Modifie [liste_bachelor]
Jonathan Malfoy(HEG)	2010-10-14 09:20:57	Connexion à l'application	**Jonathan Malfoy** s'est connecté avec succès en tant que [UTILISATEUR][ADMINISTRATEUR][UTILISATEUR]

5. Redondance

5.1 Généralité

La mise en place d'une redondance est cruciale pour le système d'alarme car si un serveur venait à être coupé du réseau, les alarmes ne pourraient plus être détectées et traitées ce qui pourrait avoir des conséquences relativement dramatiques selon la nature de l'alarme. Les serveurs de l'état de Genève sont configurés par défaut pour la redondance grâce au système HeartBeat et Unison.

5.1.1 Système HeartBeat

Heartbeat est un système de gestion de la haute disponibilité sous Linux, FreeBSD, OpenBSD,, Solaris et MacOS X. Il est distribué sous licence GPL.

Heartbeat met en place un système classique de clustering en haute disponibilité basé sur des battements de cœur. Il exécute des scripts d'initialisations lorsque une machine tombe (plus d'entente du battement de cœur) ou est à nouveau disponible (battement de cœur retrouvé). Il permet aussi de changer d'adresse IP entre les deux machines à l'aide de mécanismes ARP avancés. Heartbeat fonctionne à partir de deux machines et peut être mis en place pour des architectures réseaux plus complexes.[14]

5.1.2 Système Unison

Unison est un logiciel de synchronisation de fichiers, doté également de fonctionnalités lui permettant de créer et gérer des sauvegardes de répertoires. La synchronisation étant bidirectionnelle (en ce sens que les modifications d'un répertoire sont reportées sur l'autre et vice-versa), il est utilisé pour garder à niveau des répertoires se trouvant en même temps sur deux machines différentes.

Sur les serveurs, Unison utilise le mode distant: le programme client contacte le programme serveur, grâce au protocole SSH, ce qui permet une communication sécurisée à travers l'Internet. La version d'Unison entre le programme serveur et le programme client doit être identique[15].

Dans le cadre de l'application, le système Unison est utilisé pour synchroniser les fichiers nécessaires au déroulement de l'application comme la configuration d'Asterisk, HeartBeat ainsi que les scripts.

---

[14] Définition - http://fr.wikipedia.org/wiki/Heartbeat_(programme)

[15] Définition - http://fr.wikipedia.org/wiki/Unison_(logiciel)

5.2 *Réplication de la base de données*

Afin de garantir la haute disponibilité de la base de données, un système de réplication multi-maîtres a été mis en place. Pour éviter les conflits entre clefs, les enregistrements dans les bases de données se font, sur un serveur, en incrémentant de façon que les clefs primaire soient paires et impaires pour l'autre serveur. Le système possède, au final, deux serveurs mysql indépendant proposant un système de réplication asynchrone. Si on part du principe que les daemons tournent sur une seule machine à la fois, cela ne pose pas de problème de synchronisation mais il peut arriver que les daemons se lancent sur les deux serveurs (vois sous-chapitre suivant).

5.3 *Daemon*

En général, les daemons tournent en continu et de façon unique sur les serveurs. Mais comme ils dépendent de ressources externes, comme mysql par exemple, il se peut qu'une erreur les arrête de façon prématurée. C'est pourquoi il faut penser à mettre en place un système de redémarrage automatique. L'outil Monit a été spécialement conçu pour résoudre ce problème.

5.3.1 Monit

Monit est un outil open source de surveillance des services locaux installés sur une machine Unix. Il peut vérifier la disponibilité d'un daemon et les ressources occupées qu'il consomme, et en fonction du résultat choisir de le laisser tranquille, de le redémarrer ou de le stopper[16]. Monit journalise (avec syslog) toutes les actions prises.

Si un daemon venait à s'arrêter, Monit s'occupera de le redémarrer et évitera ainsi de rendre le système indisponible.

Si un serveur venait à tomber en panne, les daemons devront se lancer sur l'autre machine. Les conditions de lancement des daemons peuvent être configurées en utilisant un langage propre à Monit. Si le serveur redondant ne répond plus, Monit va alors lancer les daemons sur le serveur « en vie ». Lorsque le serveur tombé redémarrera, Monit devra stopper les daemons sur un des deux serveurs. L'un des deux doit avoir la priorité sur l'autre afin de savoir qui doit lancer les daemons lorsque les deux serveurs sont démarrés.

---

[16] Définition - http://fr.wikipedia.org/wiki/Monit

5.3.2 Panne réseau entre les deux serveurs

Il peut arriver qu'une panne entre les deux serveurs se produise ce qui aurait pour conséquence le blocage de toute communication provocant la perte momentanée de la redondance. La base de données ne serait alors plus répliquée. Dans ce cas-là, les daemons se lanceront sur les deux serveurs et provoqueront une désynchronisation des données présente sur la base de données. Seul une intervention manuelle de resynchronisation de la base de données peut régler le problème

Conclusion

En conclusion, on peut remarquer que les logiciels libres et open-source constituent une véritable alternative aux logiciels fermé. D'une part grâce à un support maintenu par une forte communauté à travers des forums riches et des documents d'aide complet et d'autre part grâce aux respects des normes de la VoIP.

Cependant, du fait de la récente émergence de la téléphonie par IP, certains modules ne font pas encore totalement finalisés comme, par exemple, la prise en charge de la langue française dans le module de Text-To-Speech d'Asterisk.

Pour la mise en conférence, on se rend compte que les utilisateurs ne réagissent pas forcement comme le système l'attend. On a du donc analyser et prendre en compte les différentes actions qu'un utilisateur peut effectuer lorsqu'il est mis en conférence avec plusieurs autres personnes. Par exemple, par défaut, le système attend une quittance de la part d'un appelé lors d'une mise en conférence mais la personne peut omettre ce processus.

Plusieurs autres problèmes non prévu sont apparus lors du développement de l'application comme la boite vocale qu'Asterisk prend comme une réponse à un appel ou la norme sur les touches DTMF pas encore prise en charge par tous les constructeurs de téléphone. On peut donc relever que malgré les accords pris sur les normes de diffusion par VoIP, on reste dépendant de la volonté du constructeur pour les introduire dans leurs Natels.

Ce travail a été une véritable source d'enrichissement tant au niveau de la téléphonie par IP que part les diverses étapes de développements, de l'interface web en passant par la redondance ainsi que la réalisation des daemons unix.

Bibliographie

Déon, S., *VoIP et ToIP – Asterisk – La téléphonie sur IP,* St Herblain, Editions ENI, 2007

Sultan, P., *Asterisk – La téléphonie d'entreprise libre,* Paris, Editions Eyrolles, 2009

http://www.voip-info.org – A reference guide to all things VOIP

http://www.wikipedia.org – L'encyclopédie libre

http://www.asterisk.org -- The Open-Source Telephony Project

Annexe 1
Manuel d'installation

L'installation a été testée sur le système d'exploitation **Debian 5.0.5 64 bits**

Configuration d'Asterisk 1.4

Installez les paquets suivants :

« libmime-lite-perl », requis pour l'envoi de mail

« libasterisk-agi-perl », prise en charge du langage PERL pour les scripts AGI

Fichier /etc/asterisk/extensions.conf

Ajoutez la macro suivante dans le fichier de configuration :

```
[macro-dialout_callmanager]
exten => s,1,Dial(SIP/${ARG1}@cmtr001)
exten => s,n,Dial(SIP/${ARG1}@cmtr004)
exten => s,n,Dial(SIP/${ARG1}@cmtr005)

exten => s,n,Congestion
```

Ainsi que l'extension suivante :

```
[out]
exten => _0[2-9]XXXXXXXX,1,Macro(dialout_callmanager,${EXTEN})
exten => _00[2-9]XXXXXXXX,1,Macro(dialout_callmanager,${EXTEN:1})
exten => _8XXXX,1,Macro(dialout_callmanager,02238${EXTEN})
exten => _6[1-8]XXX,1,Macro(dialout_callmanager,02254${EXTEN})
exten => _5XXXX,1,Macro(dialout_callmanager,02232${EXTEN})
exten => _7XXXX,1,Macro(dialout_callmanager,02232${EXTEN})
```

Ensuite copier les deux extensions suivantes nécessaires pour l'enregistrement audio et la mise en conférence :

```
[record_conferenceWav]
exten => s,1,Answer()
exten => s,n,Wait(2)
exten => s,n,Playback(${INTRO_RECORD})
exten => s,n,Record(${FILE_RECORD},,120)
exten => s,n,Playback(beep)
exten => s,n,Wait(2)
exten => s,n,Hangup()

[conference_alarme]
exten => s,1,Set(CHANNEL(language)=fr)
exten => s,n,Answer()
exten => s,n,Wait(2)
exten => s,n,Playback(${INTRO_CONF})
exten => s,n,WaitExten(10)
exten => #,1,AGI(logalarme.agi|${NOM_ALARME}|${LOGIN}|est entre en conference)
exten => #,n,MeetMe(${NUM_CONF},cdMFX)
exten => #,n,Hangup()
exten => *,1,AGI(quittance.agi|${ID_CRITERE})
exten => *,n,AGI(logalarme.agi|${NOM_ALARME}|${LOGIN}|a quittance)
exten => *,n,MeetMeAdmin(${NUM_CONF},K)
exten => *,n,Playback(${QUITTANCE_CONF})
exten => *,n,Hangup()
exten => t,1,Playback(${GOODBYE_CONF})
exten => t,n,Hangup()
exten =>
h,1,DeadAGI(sendmail.agi|${NOTIFICATION}|${EMAIL}|${NOM_ALARME}|${LOGIN})
exten => h,n,MeetMeCount(${NUM_CONF}|count)
exten => h,n,Gotoif,$[${count} = 0}]?103
exten => h,103,DeadAGI(end.agi|${NOM_ALARME})
```

Les mots en rouge peuvent être modifiés. Par défaut, # est utilisé pour rentrer en conférence et * pour quittancer.

Fichier */etc/asterisk/sip.conf*

S'assurer que la méthode d'envoi DTMF est bien rfc2833

```
dtmfmode=rfc2833
```

Scripts AGI

Copiez les scripts AGI (disponible dans le répertoire « agi-bin » du dossier d'installation) dans le répertoire */usr/share/asterisk/agi-bin* et mettre les droits root (chown) sur tous les fichiers.

Si besoin est, vous pouvez changer la **configuration de l'envoi de mail** dans le fichier « *sendmail.agi* » ainsi que **l'envoi de quittance** dans le fichier « *quittance.agi* » (Attention, si vous changez le port et l'adresse, il ne faudra pas oublier de le modifier aussi dans le daemon de quittance)

Installation des daemons

Installez la librairie de prise en charge mysql pour le langage c « libmysqlclient-dev »

Daemon alarme

Dans le fichier *« daemon_alarmes.c »* modifiez les constantes de connexion aux bases de données.

```
#define SERVER

#define USER

#define PASS

#define BDDALARMES

#define BDDLISTES
```

Éditez le fichier « *Makefile* » et changez la ligne de compilation selon le répertoire d'installation du daemon.

Compilez le daemon avec la commande make all.

Daemon quittance

Dans le fichier « *daemon_quittance.c* » modifiez les constantes de connexion aux bases de données.

#define SERVER

#define USER

#define PASS

#define BDD

#define PORT (port de connexion socket)

Éditez le fichier « *Makefile* » et changez la ligne de compilation selon le répertoire d'installation du daemon.

Compilez le daemon avec la commande make all.

Daemon exécute

Dans le fichier « *main.cpp* » modifiez les constantes de connexion aux bases de données.

#define SERVER

#define USER

#define PASS

#define BDD

#define CHEMIN (répertoire d'exécution des scripts)

#define CHEMINSIMPLE (pour tester l'existence des scripts)

Compilez le daemon avec la commande

g++ main.cpp –lmysqlcient –o daemon_execute

Daemon critère

Dans le fichier « *main.cpp* » modifiez les constantes de connexion aux bases de données.

```
#define SERVER

#define USER

#define PASS

#define BDD
```

Compilez le daemon avec la commande

```
g++ main.cpp –lmysqlcient –o daemon_critere
```

Monit

Ajoutez les lignes suivantes dans le fichier de configuration /etc/monit/monitrc sur les deux serveurs :

```
check process heartbeat with pidfile /var/run/heartbeat.pid
      .start program = "/etc/init.d/heartbeat start"
       stop  program = "/etc/init.d/heartbeat stop"
       depends asterisk, mysql

check process asterisk with pidfile /var/run/asterisk/asterisk.pid
       start program = "/etc/init.d/asterisk start"
       stop program  = "/etc/init.d/asterisk stop"
       if failed port 5060 type udp then restart

check process mysql with pidfile /var/run/mysqld/mysqld.pid
       start program = "/etc/init.d/mysql start"
       stop program = "/etc/init.d/mysql stop"
       if failed port 3306 protocol mysql then restart
```

Puis ces lignes sur le serveur principal (un pour chaque daemon):

```
check process service with pidfile /var/run/nom_du_daemon.pid
        start program = "/etc/init.d/nom_du_daemon"
      stop program = "/sbin/start-stop-daemon –stop –pidfile
/var/run/nom_du_daemon.pid "
        if failed port 80 protocol tcp then start
```

Changez les mots en rouge pour chaque daemon et selon le répertoire d'installation des daemons.

Attention, le daemon_alarme doit impérativement se lancer en root (sudo) du à une contrainte Asterisk sur les fichiers d'appel.

Répétez l'opération sur l'autre serveur mais en changeant la dernière ligne :

```
check process service with pidfile /var/run/nom_du_daemon.pid
        start program = "/etc/init.d/nom_du_daemon"
       stop program = " /sbin/start-stop-daemon –stop --pidfile /var/run/
nom_du_daemon.pid "
        if failed host ip_du_serveur_principal port 80 protocol tcp then start
        if host ip_du_serveur_principal port 80 protocol tcp then stop
```

Interface de gestion

Copier les répertoires gest_listes et gest_test vers le répertoire /var/www/

Exécutez (en root) les scripts SQL contenu dans le répertoire /sql du dossier d'installation.

Lancement des daemons

Utilisez la commande **nohup** pour lancer les daemons. Exemple :

```
nohup ./daemon_quittance &
```

Annexe 2
Manuel utilisateur

Authentification

En entrant l'adresse internet de l'application, vous allez devoir choisir entre une authentification par **GINA** ou **LDAP.** Une fois l'authentification effectuée, vous allez être redirigé sur l'interface de gestion.

Résumé des droits utilisateur selon leurs rôles

	Gestions des utilisateurs	**Gestion des listes d'appels**	**Gestion des conférences**	**Gestions des textes à jouer**
Consultation	Administrateur global, Utilisateur*	Administrateur global, Administrateur de la liste, Utilisateur**	Administrateur global	Administrateur global
Ajout	Administrateur global	Administrateur global	Administrateur global	Administrateur global
Modification	Administrateur global, Utilisateur*	Administrateur global, Administrateur de la liste, Utilisateur**	Administrateur global	Administrateur global
Suppression	Administrateur global	Administrateur global	Administrateur global	Administrateur global

* : L'utilisateur peut consulter et modifier seulement son profil.

** : L'utilisateur peut consulter seulement les listes auxquels il appartient et modifier seulement son inscription et sa notification

Utilisateurs

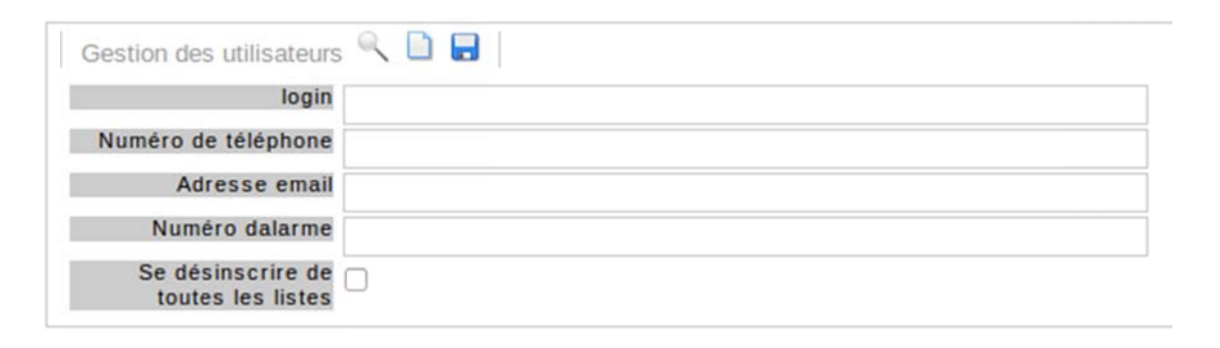
Création d'un utilisateur (Administrateur global seulement)

Un profil utilisateur est créé à chaque première connexion. Cependant, il peut être utile de créer manuellement un utilisateur.

1) Cliquez sur l'onglet « Gestion des utilisateurs »

2) Cliquez sur l'image d'ajout

3) Vous pouvez maintenant remplir les champs correspondants :



: Revenir à la liste des utilisateurs

: Effacer les champs

: Sauver l'utilisateur

Modification d'un utilisateur

L'utilisateur peut modifier ses informations personnelles et l'administrateur global peut effectuer des modifications sur tous les utilisateurs.

1) Cliquez sur l'onglet « Gestion des utilisateurs »

2) Cliquez sur l'icône de modification pour l'utilisateur concerné (dans la colonne « action »)

3) Sauvegardez les modifications en cliquant sur l'icône

Suppression d'un utilisateur (Administrateur global seulement)

1) Cliquez sur l'onglet « Gestion des utilisateurs »

2) Cliquez sur l'icône de suppression pour l'utilisateur concerné (dans la colonne « action »)

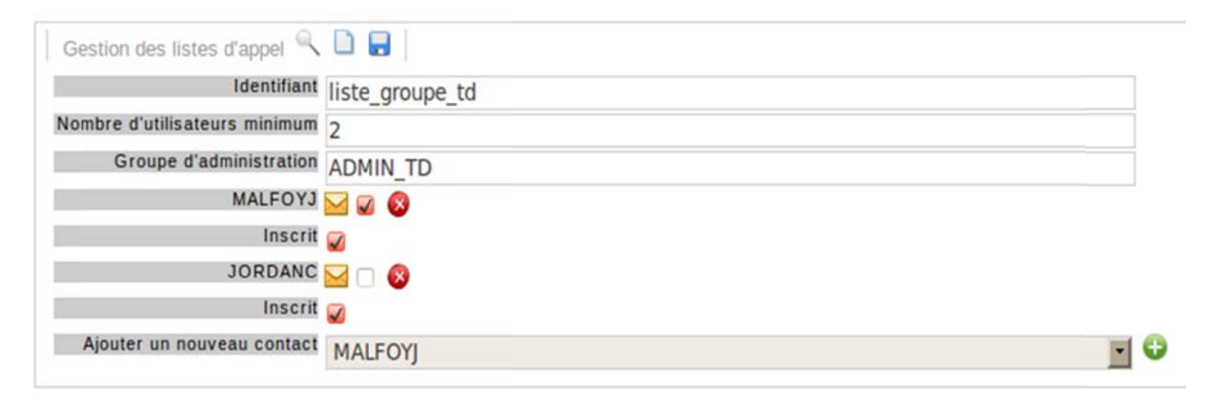
Liste d'appel

Création d'une liste d'appel (Administrateur global seulement)

1) Cliquez sur l'onglet « Gestion des listes d'appel »

2) Cliquez sur l'image d'ajout

3) Vous pouvez maintenant remplir les champs correspondants :



Remarque : Tous les utilisateurs faisant partie du groupe d'administration inscrit dans le champ correspondant peuvent modifier la liste (mais pas la supprimer)

: Revenir à la liste

: Effacer les champs

: Sauver la liste d'appel

Pour ajouter un utilisateur, cliquez sur l'icône après avoir choisi l'utilisateur dans le champ « Ajouter un nouveau contact »

Ensuite pour chaque utilisateur :

: Notifie par mail l'utilisateur associé

: Supprime l'utilisateur de la liste d'appel

Inscrit : Inscrit l'utilisateur pour la réception d'appel

Modification d’une liste d’appel

Chaque utilisateur peut choisir s’il veut être notifié et inscrit dans une liste d’appel. Il doit avoir été préalablement inscrit dans cette liste.

L’administrateur de la liste peut modifier les listes associé à son groupe d’administration

L’administrateur global possède tous les droits sur tous les champs.

1) Cliquez sur l’onglet « Gestion des listes d’appel »

2) Cliquez sur l’icône de modification pour la liste d’appel concernée (dans la colonne « action »)

3) Sauvegardez les modifications en cliquant sur l’icône

Suppression d’une liste d’appel (Administrateur global seulement)

1) Cliquez sur l’onglet « Gestion des listes d’appel »

2) Cliquez sur l’icône de suppression pour la liste d’appel concernée (dans la colonne « action »)

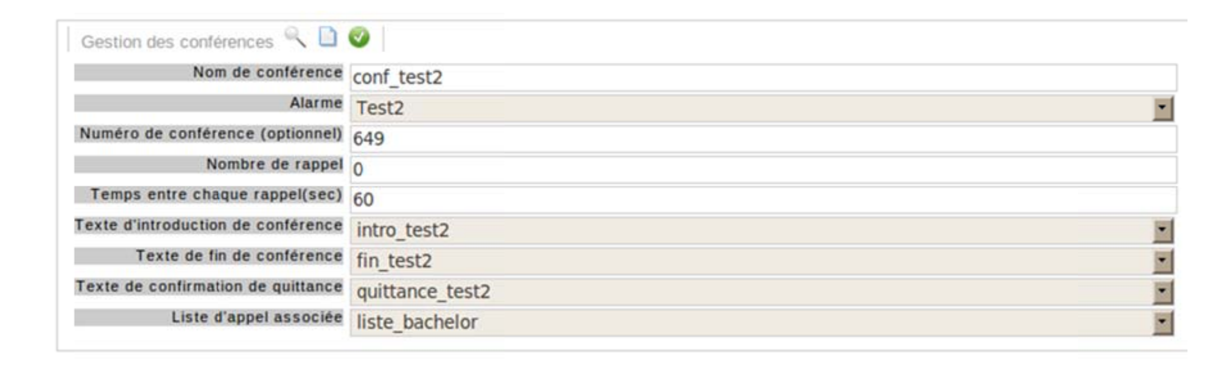
Conférence

Création d'une conférence (Administrateur global seulement)

1) Cliquez sur l'onglet « Gestion des conférences »

2) Cliquez sur l'image d'ajout

3) Vous pouvez maintenant remplir les champs correspondants :



Le **numéro de conférence** est optionnel. Si le champ est vide, l'application générera un numéro aléatoire.

Le **nombre de rappel** ainsi que le **temps entre chaque rappel** correspondent au nombre de fois qu'Asterisk va appeler l'utilisateur si celui ne décroche pas.

Remarque : Lorsqu'un appel tombe sur une boite vocale, Asterisk le prend en compte comme décroché.

: Revenir à la liste des conférences

: Effacer les champs

: Sauver la conférence

Modification d'une conférence (Administrateur global seulement)

L'administrateur global peut modifier tous les champs.

1) Cliquez sur l'onglet « Gestion des conférences »

2) Cliquez sur l'icône de modification pour la conférence concernée (dans la colonne « action »)

3) Sauvegardez les modifications en cliquant sur l'icône

Suppression d'une conférence (Administrateur global seulement)

1) Cliquez sur l'onglet « Gestion des conférences »

2) Cliquez sur l'icône de suppression pour la conférence concernée (dans la colonne « action »)

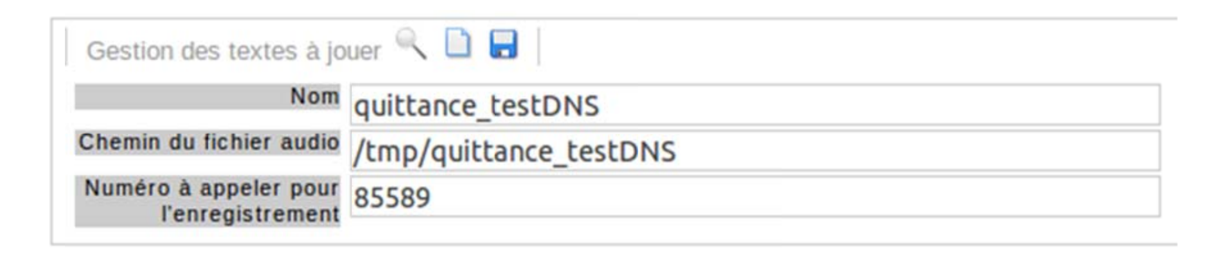
Texte à jouer

Création d'un texte à jouer (Administrateur global seulement)

1) Cliquez sur l'onglet « Gestion des textes à jouer »

2) Cliquez sur l'image d'ajout

3) Vous pouvez maintenant remplir les champs correspondants :



Le **chemin du fichier audio** ne doit pas contenir d'extension (.wav, .mp3), Asterisk gère tout seul le format du fichier.

: Revenir à la liste des textes à jouer

: Effacer les champs

: Sauver le texte à jouer.

Une fois le texte à jouer enregistré, vous recevrez **immédiatement** un appel pour effectuer l'enregistrement audio.

Modification d’un texte à jouer (Administrateur global seulement)

L’administrateur global peut modifier tous les champs.

1) Cliquez sur l’onglet « Gestion des textes à jouer »

2) Cliquez sur l’icône de modification pour le texte à jouer correspondant (dans la colonne « action »)

3) Sauvegardez les modifications en cliquant sur l’icône

Suppression d’un texte à jouer (Administrateur global seulement)

1) Cliquez sur l’onglet « Gestion des textes à jouer »

2) Cliquez sur l’icône de suppression pour le texte à jouer correspondant (dans la colonne « action »)